WHAT IS CLAIMED IS:

1. A circuit adapted to recover useful data from a video packet that is at least partially corrupted, the circuit comprising:

a decoding circuit configured to decode a video packet in a forward direction and in a backward direction, where the decoding circuit detects bit locations of errors first encountered in the forward direction and in the backward direction;

a counter adapted to maintain a count of complete macroblocks decoded in the forward direction and in the backward direction; and

a control circuit adapted to discard at least a portion of the video packet that corresponds to an overlapping region, where the control circuit is further configured to further discard additional data corresponding to a backtracking amount when there is no overlapping region, where the control circuit is further configured to discard information in incomplete macroblocks, and where the control circuit is adapted to permit use of at least a portion of the remaining data.

2. The circuit as defined in Claim 1, wherein the backtracking amount is 90 bits in each direction.

3. The circuit as defined in Claim 1, wherein the control circuit is further configured to discard an intra-coded macroblock from a partially corrupted video packet.

4. The circuit as defined in Claim 1, further comprising a ring buffer, which stores a video bitstream and is coupled to the decoding circuit such that the decoding circuit can access a video packet in both the forward direction and in the backward direction.

5. A circuit adapted to recover useful data from a video packet that is at least partially corrupted, the circuit comprising:

a data parsing circuit adapted to determine whether a video packet is encoded with data partitioning enabled;

an error checking circuit configured to determine whether an error exists ahead of a motion marker of the video packet; and

a decoder coupled to the data parsing circuit and to the error checking circuit, where the decoder is adapted to decode at least a portion of the data in the corrupted video packet ahead of the motion marker when data parsing circuit indicates that the

video packet is encoded with data partitioning enabled and when the error checking circuit indicates that the error does not exist ahead of the motion marker.

6. The circuit as defined in Claim 5, wherein the error checking circuit further comprises:

a prediction circuit that generates a predicted location for the motion marker;

a circuit that detects an actual location of the motion marker; and

a comparison circuit that indicates that the error exists ahead of the motion marker when the actual location and the predicted location do not match.

7. A circuit adapted to recover useful data from a video packet that is at least partially corrupted, the circuit comprising:

means for receiving the video packet;

means for ending without recovering data when corruption is detected in at least one of a video packet header of the video packet, a DC portion of the video packet, and a motion vector portion of the video packet;

means for initiating decoding of the video packet in a forward direction;

means for maintaining a first count of a number of macroblocks decoded without error in the forward direction;

means for storing codewords decoded in the forward direction;

means for storing a first bit location when an error is first detected in the forward direction;

means for initiating decoding of the video packet in a reverse direction;

means for maintaining a second count of a number of macroblocks decoded without error in the reverse direction;

means for storing codewords decoded in the reverse direction;

means for storing a second bit location when an error is first detected in the reverse direction;

means for determining if there is an overlapping region, where the overlapping region corresponds to a region identified in both the forward direction and in the reverse direction as having an error;

means for discarding the data in the overlapping region and for using the data in a remaining portion of the video packet if there is an overlapping region; and

means for discarding the data between a first backtracking amount ahead of the first error location in the forward direction and a second backtracking amount behind the second error location in the first location, and for recovering the remaining portion of the video packet if there is no overlapping region.

8. A method of recovering useful data from a video packet that has been corrupted, the method comprising:

receiving the video packet;

ending without recovering data when corruption is detected in a video packet header of the video packet;

ending without recovering data when corruption is detected in a DC portion of the video packet;

ending without recovering data when corruption is detected in a motion vector portion of the video packet;

initiating decoding of the video packet in a forward direction;

maintaining a first count of a number of macroblocks decoded without error in the forward direction;

storing codewords decoded in the forward direction;

storing a first bit location when an error is first detected in the forward direction;

initiating decoding of the video packet in a reverse direction;

maintaining a second count of a number of macroblocks decoded without error in the reverse direction;

storing codewords decoded in the reverse direction;

storing a second bit location when an error is first detected in the reverse direction;

determining if there is an overlapping region, where the overlapping region corresponds to a region identified in both the forward direction and in the reverse direction as having an error;

if there is an overlapping region, discarding the data in the overlapping region and using the data in a remaining portion of the video packet; and

if there is no overlapping region, discarding the data between a first backtracking amount ahead of the first error location in the forward direction and a second backtracking amount behind the second error location in the first location, and recovering the remaining portion of the video packet.

9. The method as defined in Claim 8, wherein the first error location and the second error location correspond to bit locations.

10. The method as defined in Claim 8, wherein the first error location and the second error location correspond to macroblock boundaries.

11. The method as defined in Claim 8, wherein the first backtracking amount and the second backtracking amount are each to a next valid macroblock boundary.

12. The method as defined in Claim 8, wherein the first backtracking amount and the second backtracking amount are about 90 bits.

13. The method as defined in Claim 8, further comprising discarding recovered data from a corrupted video packet that corresponds to an intra-coded macroblock.

14. The method as defined in Claim 8, further comprising:

determining whether AC prediction was disabled by the encoder;

using a recovered intra-coded macroblock if the intra-coded macroblock is recovered from a portion of the video packet that is ahead of a DC marker in the forward direction, where the video packet was encoded with AC prediction disabled; and

otherwise discarding recovered data corresponding to an intra-coded macroblock.

15. The method as defined in Claim 8, further comprising using recovered data corresponding to a first intra-coded macroblock only if no other intra-coded macroblock exists to the immediate left of the first intra-coded macroblock and no other intra-coded macroblock exists immediately above the first intra-coded macroblock in the image.

16. The method as defined in Claim 8, further comprising concealing errors with gray pixels for portions of the video packet that were not recoverable.

17. A method for recovering data in a corrupted video packet comprising:

inspecting the video packet to determine whether the video packet was encoded with data partitioning enabled;

determining whether an error exists ahead of a motion marker of the video packet; and

decoding at least a portion of the data in the corrupted video packet ahead of the motion marker when the video packet was encoded with data partitioning enabled and when the error does not exist ahead of the motion marker.

18. The method as defined in Claim 17, wherein the determining whether the error exists ahead of the motion marker further comprises:

predicting a location for the motion marker;

detecting the motion marker;

comparing an actual location of the motion marker to the predicted location of the motion marker;

determining that the error exists behind the motion marker when the actual location and the predicted location match; and

determining that the error exists ahead of the motion marker when the actual location and the predicted location do not match.

19. The method as defined in Claim 17, wherein the portion of the data decoded includes decoding of motion vectors.

20. The method as defined in Claim 17, wherein the portion of the data decoded includes decoding of not-coded macroblock flags.

21. The method as defined in Claim 17, wherein the portion of the data decoded includes decoding of luminance (DC) information.